Lecture #15 – SED (Chapter 5)

- Background

Stands for Stream Editor (batch or non-interactive)
Allows you to perform the same kind of editing tasks as those in vi and ex

By default, it copies lines from the file-list to its stdout while performing edits

- Command line options

```
sed [-n] –f script-file [file-list]
sed [-n] script [file-list]
```

-f	file (use script-file for commands)
-n	no print (do not copy lines to its stdout except when overridden by print “p”)

Examples:

```
# Print all lines containing John (i.e. grep John file1.txt)
$ sed –n ‘/John/p’ file1.txt

#  Run the sed script myscript against the file homework5
$ sed –f myscript homework5
```

- Processing

[address[,address]] instruction [arg-list]

Addresses are optional (if omitted, sed processes all lines)
Instruction is the editing command that modifies the text
Arg-list is dependent on the instruction

1. Read one line from the input file (file-list)
2. Read the first command from script file (or command line). If the address selects the input line, perform the edit on the line
3. Read the next command, and perform edit on the current version of the line
4. Repeat step 3 until all commands are exhausted
5. Go to step 1, and read another input line.

- Addresses

Can have zero, one, or two addresses in a command
Zero addresses means to process all lines
One address selects lines that match that address
Two addresses selects a range of lines

$ is the last line

Examples:	1	Means line 1
	1,$	Means all lines
	1,/^RICH/	From line 1 until a line that starts with RICH

- Instructions

d	delete
y	transform
n	next (write out currently selected line, and read next line)
a	append (appends one or more lines to the currently selected line

```
[address]a\
text \
text \
text
```

i	insert (identical to append except it places new text before line
c	change (change selected lines so they contain new text – similar to insert)
s	substitute

[address,[address]] s/pattern/replacement/[g][p][w file]

g	global flag (replace all occurrences on the line)
p	print flag (overrides –n option)
w	sends ouput to a specified file

p	print (writes selected line to stdout overriding –n option
w	write (similar to print except to a file instead of stdout)
r	read (read contents of a file, and append to selected line)
q	quit (cause sed to stop processing)
!	NOT
{}	grouping

- Print and Quit

```
$  cat new
1	Line one.
2	The second line.
3	The third.
4	This is line four.
5	Five.
6	This is the sixth sentence.
7	This is line seven.
8	Eighth and last.

#  Print all lines from file new, and repeat lines containing “line”
$ sed ‘/line/p’ new
line 1, 2, 2, 3, 4, 4, 5, 6, 7, 7, 8

#  Print only the lines from file new containing “line”
$ sed –n ‘/line/p’ new
line 2, 4, 7

#  Print only lines 3 through 6
$ sed –n ‘3,6p’ new
3, 4, 5, 6

#  Print lines 1-5 (i.e. quit after line 5)
$ sed ‘5q’ new
1, 2, 3, 4, 5

#  Print blocks between lines that start with “1” and lines that start with “2”
$  sed –n ‘/^1/,/^2/p’ datafile
```

- Delete

```
#  Delete from line3 until the end of the file
$  sed ‘3,$d’ new
```

- Append

```
$  cat append_demo
2a \
AFTER.

$ sed –f append_demo new
1, 2
AFTER
3, 4, 5, 6, 7, 8
```

- Insert

```
$ cat insert_demo
/This/ I \
BEFORE.
1, 2, 3
BEFORE.
4, 5
BEFORE.
6
BEFORE.
7, 8
```

- Substitute

```
#  Change all occurrences of west to north
$  sed ‘s/west/north/g’ datafile

#  Change first occurrence of Monday to Tuesday on lines 1 to 1000
$  sed ‘1,1000s/Monday/Tuesday’ datafile
```

- Read and write

```
#  Insert contents of newfile after every line containing Susan
$  sed ‘/Susan/r newfile’ datafile

#  Write lines containing north to file named newfile
$  sed –n ‘/north/w newfile’ datafile
```

- Change

```
#  Replace lines containing “eastern”
$  sed ‘/eastern/c\
  THE EASTERN REGION HAS BEEN CLOSED’ datafile
```

- Transform

```
#  Uppercase contents of file
$  sed ‘y/abcdefghijklmnopqrstuvwyxz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/’ file
```

- Compound example

```
$  cat compound.in

1.        The words on this page…
2.        The words on this page…
3.        The words on this page…
4.        The words on this page…

$  cat compund
1,3 s/words/text/
2,4 s/text/TEXT/
3 d

$  sed –f compound compound.in
1.  The text on this page…
2.  The TEXT on this page…
4.  The words on this page…
```

- Modifying files with sed

  Sed is a non-destructive editor (i.e. the new version of the file is written to stdout)

```
$  sed ‘1,3d’ filex > tmp
$  mv temp filex
```